INTERNATIONAL JOURNAL OF TECHNOLOGICAL EXPLORATION AND LEARNING (IJTEL)
www.ijtel.org

Courseware Quality Groups Tie-in to Reinforcing

M.Nagarani, G.Nirali Reddy
Assistant Professor, Department of Computer Science &Engineering
Gnyana Saraswati College of Engineering & Technology
Nizamabad. Andhra Pradesh, India.

***Abstract*— Software Quality Assurance (SQA) involves the complete software development process - monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that the problems are found and dealt with. It's major aim is towards anticipation and to result in the making of quality software. This paper emphasizes the importance of a quality process and also discusses about the ways in which it could be achieved. It looks at the different ways the quality group's purpose and an enforce were viewed. The potential benefits and detracts for various authorities are discussed, along with the organizational structure and typical activities for each. The idea that the authority for the quality group changes over time is also presented, along with observed developments in the organizations. The various possible organizations, charters, and portrayal are described and related briefly to quality systems described in both the SEI Maturity Model and ISO 9000 Standards (ISO 9001 and ISO 9000-3). It describes the impact on product quality of the different types of development process, and possible roles for the software quality group.**

Keywords- Anticipation; Making; Enforce; Detracts, Portrayal.

I. INTRODUCTION

"Software Quality Assurance (SQA) involves the entire software development PROCESS - monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented towards prevention Software Quality Assurance is aimed at developing a sound software development methodology that will produce quality software. The various tasks of the software quality group are described, and the order they typically appear as the organizations grow and mature presented. This provides a foundation for understanding the contribution of the quality assurance organization and the value they can add to the product and process quality. The roles of software quality assurance basically correspond with the tasks they accomplish. The roles range from acting as an extension of development for debugging software products, to development process definition and control. Verification and validation, acceptance testing, measurement and metrics, and process consulting are also roles that software quality groups sometimes assume. The various charters that the organization may assume are described, and the impact on quality is addressed for each charter.

As the organizations grow and change, the needs and roles also change. Depending on the type of product and organization itself, the life cycles may differ and the tasks done by the quality organization evolve. The evolution takes familiar tracks, following patterns based upon the maturity of the organization and other factors. The SEI Maturity Model and other standards are relevant in uality group.

II. SIGNIFICANCE OF SQA

There is an increasing use of software, in all walks of life. From electronic devices like watches, and cell phones to applications like ecommerce, banking, medical and what not? Computer Systems are omnipresent and all computers run some software. So, software is omnipresent. Due to the widespread acceptance, and use of software systems, in various areas, software bugs are proving to be costly, and sometimes fatal. The Sustainable Computing Consortium, a collaboration of major corporate IT users, university researchers and government agencies, estimates that buggy or flawed software cost businesses $175 billion worldwide in 2001 Interested readers are referred for a list of some of the recent, major computer system failures, caused by software bugs, and its consequences. Bugs have affected banking systems, stock exchanges, medical institutions, educational institutions and even the Social Security Administration. Most bugs, encountered during software development, can be avoided, by adopting a sound software development process, and having strict software quality control using Software Quality Assurance. The process of SQA is comparable to Software Testing.

III. QUALITY GROUP PURPOSE AND ROLES

Below table 1 shows the basic purposes, roles, and activities established for software quality groups. It also describes some of the activities and roles for such organizations. Although an organization my fit the description well at one time, it is likely to change as the organization evolves and matures. It is also likely that any given quality group has characteristics of many of the organizations. Generally, however, there is one primary or predominant theme in the group. When the goal of the organization is to test products, the group usually acts as an extension of the development organization in performing a debugging function. The group's primary activities are to develop and run tests, with the primary emphasis on reporting defects. The majority of time is generally spent running the tests and reporting the results.

TABLE I. QUALITY GROUP GOALS AND ACTIVITIES

Objective	**Exertion**	**Portrayal**
Test Products	Test development, test execution	Testers; extension of development
Measure Products	Test oversight, reporting results	Measurers; Quality hurdle

Measure Processes	Metrics	Information Engineers
Define Processes	Process and Risk management	Quality and Process Engineers
Guidance Resource	Quality Reference	Quality Engineers

TABLE II. MOST APPROPRIATE SOFTWARE DEVELOPMENT TECHNIQUES BASED UPON PRODUCT REQUIREMENTS

Life Cycle	Product Requirements
Waterfall	Known, unchanging
Prototyping	Unknown, changing
Spiral Unknown,	unchanging
Decomposition/Integration	Known, unchanging
Cleanroom	Known, provable
Fourth Generation Techniques	Unknown

A different goal for the quality group occurs when they focus more on the organization's processes, rather than the products. This often occurs when the organization focuses on metrics programs and expands beyond the nebulous "defects happen" theory into an understanding that "defects are built in". The role becomes more general, that of information brokers seeking insights from whatever data is obtainable. These metrics programs change the role of the quality group in information engineers; applying the data to understand and improve the organization.

IV. SOFTWARE DEVELOPMENT LIFE CYCLES

The various Software Development Organization Life Cycles (SDLC) are described in Table 2, and situations where they are most applicable and effective are shown. The SDLCs described include the classic waterfall, prototyping, spiral, decomposition/integration, and variations encountered and created in various organizations. The appropriateness for the life cycles is described in relation to the stability and understanding of product requirements.

For example, the classic waterfall approach to software development is most appropriate when the requirements can be fully known before beginning development, and they don't Change substantially during the product development. If they change substantially, a spiral approach is more likely to fit the organization's needs.The quality group's roles occur independently of the life cycle involved. The specific activities differ on a technical level, but the various possible roles remain the same, and the progression and evolution occur in the same ways.

V. ORGANIZATION MATURITY

Organization maturity is not an indication of the age of the group. It has been defined as a loose measure of the formality of the processes used by software development. In my experience, this maturity roughly correlates to the role of the software group.

The relationship of the charter of the software quality assurance group to SEI's Process Maturity Model is shown in Table 3. The five levels of maturity generally occur in organizations with specific roles defined for the software quality groups. This close relationship between the organization maturity and the role of the software quality group is worthwhile understanding. Although they seem closely correlated, I believe there is a chicken-and-egg problem in trying to determine which causes which.

The role of the quality group evolves from testing to process definition and control as the

Organization evolves. Trying to control and optimize the development process in an organization at the Initial Level does not make sense. On the other hand, paying no attention to process does not make sense either.

The most effective role for the quality group is the one that best supports the organization today, while preparing to improve it in the near term. Without advocating any particular model For organization development, the quality group must understand and support some model -Whatever model the organization agrees fits its needs.

TABLE III. ORGANIZATION MATURITY AND SQA ROLES

SEI Maturity Level	Role of Software Quality Assurance
Initial	Testing
Repeatable	Quality hurdle
Defined	Oversight, Metrics
Managed	Process and Risk management
Optimizing	Reference, Oversight

Other models and standards, such as ISO 9000, may also be applied. The role of ISO 9000 is as the framework for a quality system, rather than a process methodology or prescription for the software quality organization's charter or function. The relationship of the quality system to the business system and development methodology is graphically described in Table 4.

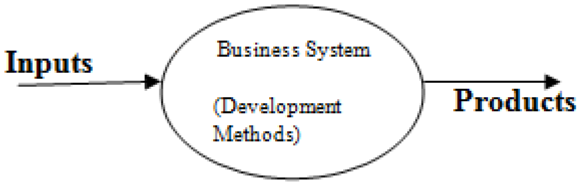

Figure 1. Quality System and Business System Relationship

Neither SEI's Process Maturity Model nor ISO 9000 describe in detail what the right process is, who should do what, or how things should be done. ISO does not begin to prescribe these things, but rather provides rules for knowing if a given quality system might qualify under its guidelines. Neither system addresses the product or product quality directly.

The models don't prescribe methods because each problem situation is different to the point that there is no unique solution. In order for generalized models for development organizations or quality systems to be useful, they must be applicable in many situations. If the models prescribed specific methods and techniques, they would not be applicable to the majority of organizations that have different needs and characteristics. Organizational requirements are unique, and are based upon the product characteristics, customer needs, and organizational politics.

The models are also process oriented, not product oriented. They focus on the processes organizations should have, not the products. They do not address testing of products or product quality directly. They point out how the process must be defined, controlled, and improved. Only by controlling the process can the product quality be predicted and controlled.

VI. SOFTWARE QUALITY ASSURANCE VS SOFTWARE TESTING

Software Testing involves operating a system, or an application, under controlled conditions, and evaluating the results. In most cases, software testing will involve the development of a test bed, which tests the given software, upon a set of test cases. The test bed will feed the test input to the software system, get the result that's generated by the software system, and compares the generated result with the expected result. If the generated result is same as the expected result, then the software is bug free else, it has bugs that need to be fixed.

Software testing is normally carried out under controlled conditions. The controlled conditions should include both normal and abnormal conditions. The aim of testing is to try to break the software, and find the bugs in it. Successful testing will discover all the bugs in the software. Developing automated test tools to perform testing is an active area of research. Testing is oriented towards *'detection'* of bugs in the software (An interesting article that discusses about how extensive testing should be can be found in [4]). On the other hand, SQA is aimed at avoiding bugs.

Software Quality Assurance is more challenging than Software Testing because, solving problems is a high-visibility process; preventing problems is a low-visibility process. During Software Testing, we know what the problem is, and we are trying to fix the problem, which is easier than, preventing the problem before it occurred, or even showed signs of occurrence.

A. *Software Testing Maturity Model*

Why should we assess the testing maturity?

The testing maturity models were developed around 1996. SW-TMM is a testing process improvement tool that can be used either in conjunction with the SW-CMM or as a stand-alone tool

There is no consistency within their organization as to the health and professionalism of the testing process. An assessment of the testing process using a testing maturity model will document the current level. Highlight the variances between the imagined level and the actual level. Provide a road map for making the necessary process improvements

B. *Testing maturity models*

They are different types of testing maturity models

- Testability Maturity Model.
- Software Testing Maturity Model.
- Test Process Improvement.
- Test Organization Maturity TM.
- Testing Assessment Program.
- Proposed Evaluation and Test SW-CMM Key Process Area

C. *Software Testing Maturity Model*

The five levels of SW-TMM

- Level 1: Initial
- Level 2: Phase Definition
- Level 3: Integration
- Level 4: Management and Measurement
- Level 5: Optimization/Defect Prevention and Quality Control

1) *Software Testing Maturity Model Level 1*

In this level it is a chaotic process and it is not distinguished from debugging and ill defined where as the tests are developed ad hoc after coding is complete and this is usualy lack a trained professional testing staff and testing tools and the main objective of testing is to show that the system and software work

2) *Software Testing Maturity Model Level 2*

In this level we Identify testing as a separate function from debugging where as testing became a defined phase following coding. The main objective of testing is to show that the system and software meets specifications are reached.

3) *Software Testing Maturity Model Level 3*

In this level an integrate testing is entered into the entire life cycle to establish a formal testing organization, establishes formal testing technical trainings and which controls and monitors the testing process to begins to consider using automated test tools. The main objective of testing is based on system requirements. Major milestone reached at this level: management Recognizes testing as a professional activity

4) *Software Testing Maturity Model Level 4*

In this level the testing is a measured and quantified process and the Development products are now tested for quality attributes such as Reliability, Usability, and Maintainability. Test cases are collected and recorded in a test database for reuse and regression testing

Defects:

- Now logged,
- Given a severity level,
- assigned a priority for correction

5) *Software Testing Maturity Model Level 5*

In this level the testing is institutionalized within the organization and the Testing process is well defined and managed. Testing costs and effectiveness are monitored. Automated tools are a primary part of the testing process. There is an established procedure for selecting and evaluating testing tools

Characteristics of SW-TMM

- It is easy to understand and use
- It provide a methodology to baseline the current test process maturity
- It is designed to guide organization
- It is used for selecting process improvement strategies
- It is used for identifying critical issues to test process maturity
- It provide a road map for continuous test process improvement
- It provide a method for measuring progress
- It allow organizations to perform their own assessment

VII. Conclusion

The role of the quality group should be set based upon the needs of the organization. These needs can be predicted by the maturity of the organization and the need to change. Then goals for improvement of the process and evolution of the organization can be set. The quality group can play a big part in the planning and implementation through understanding of organizational development needs and techniques. Then an improvement program to attain the goals can be begun. This is the foundation of any continuous improvement program, and ultimately should be the goal of the software quality group and all of management in the organization.

References

[1] Humphrey, W.S., *Managing the Software Process,* Addison-Wesley, 1989

[2] Humphrey, W., *"Characterizing the Software Process: A Maturity Framework,"* IEEE

Software (March), 73-79, 1988.

[3] *Quality Management and Quality Assurance Standards,* ASQC, documents ANSI/ASQC Q90, Q91, Q92, Q93, and Q94-1987.